REMARKS

Claims 1-9, 11-15, 17-22, 24-34, 36-43, 45, and 51 were pending in the application, with Claims 1, 12, 19, 26, and 38 being independent. Applicant amends Claims 1, 12, 19-26, and 38 to further clarify features of the claimed subject matter. The original specification and drawings support these claim amendments at least at pages 2, 4, 8, 11-13, 18, 20-21, and 28, and in Figures 1, 5b, and 6. These revisions introduce no new matter.

Claims 1-9, 11-15, 17-22, 24-34, 36-43, 45, and 51 are now pending in the application. Applicant respectfully requests reconsideration and allowance of the subject application in view of the foregoing amendments and the following remarks.

Claim Rejections Under 35 U.S.C. § 101

Claims 1-9, 11-15, 17, 19-22, and 24-25 stand rejected under 35 U.S.C. § 101 as being directed to non-statutory subject matter. Specifically, Claims 1-9, 11-15, and 17 are rejected because they are directed to a method for estimating security requirements, and Claims 19-22 and 24-25 are rejected because they are directed to a computer-readable storage which may comprise "computer storage media" and "communications media". Applicant respectfully traverses the rejection.

Applicant amends independent **Claims 1 and 12** to recite a "*method implemented on a computing device having instructions stored on a computer-readable storage media and executable by a processor, . . . comprising:*", as discussed during the interview. The support for these amendments may be found at least on pages 18 and 20, and in Figures 1 and 6. Thus, no new matter has been introduced.

Applicant amends independent **Claim 19** to recite "*one or more computer storage media having a tangible component comprising instructions that, when executed by a processor, perform* . . .". The support for this amendment may be found at least on pages 18 and 20-21, and in Figures 1 and 6. Thus, no new matter has been introduced.

Dependent Claims 2-9, 11, 13-15, 17, 20-22, and 24-25 depend directly or indirectly from one of independent Claims 1, 12, or 19, respectively, and thus are allowable as depending from an allowable base claim. Furthermore, The Examiner and the Supervisor tentatively agreed during the interview that the proposed amendments overcome the § 101 rejections.

Applicant respectfully submits that these claims are no longer directed to non-statutory subject matter and respectfully requests that the § 101 rejections be withdrawn.

Claim Rejections Under 35 U.S.C. § 112, second paragraph

Claims 1-9, 12-15, and 17 stand rejected under 35 U.S.C. § 112, second paragraph, as being indefinite for failing to particularly point out and distinctly claim the subject matter which applicant regards as the invention. Specifically, the Office Action rejects these claims because the independent Claims 1 and 12 claim two methods in the same claim (Office Action, pg. 4). Applicant respectfully traverses the rejection.

Applicant amends **independent Claims 1 and 12** to clarify the distinction between two methods recited. For example, Claims 1 and 12 as amended recite "*gathering a permission set for the* ***method in the assembly****; determining whether the* ***method in the assembly*** *calls* ***another method in the assembly*** *or in an another assembly*". The support for this amendment is found at least at pages 8, 11-12, and 16.

Thus, no new matter has been introduced. Applicant respectfully submits that these claims are no longer indefinite, and accordingly, Applicant respectfully requests withdrawal of § 112 rejection for these claims.

Dependent Claims 2-9, 13-15, and 17 depend directly or indirectly from one of independent Claims 1 or 12, respectively, and thus are allowable as depending from an allowable base claim. Applicant respectfully submits that these claims are no longer indefinite and respectfully requests that the § 112 rejections be withdrawn.

Claim Rejections Under 35 U.S.C. § 103(a)

Claims 1-9, 11-15, 17-22, 24-34, 36-43, 45, and 51 stand rejected under 35 U.S.C. § 103(a) as being unpatentable over Reference U, *IBM Research Report: Access Rights Analysis for Java*, by Koved et al. (hereinafter "Koved"), in view of U.S. Patent Application Publication No. 2002/0174224 to Scheifler et al. (hereinafter "Scheifler"). Applicant respectfully traverses the rejection.

Independent Claim 1

Without conceding the propriety of the stated rejections, and only to advance the prosecution of this application, Applicant amends independent Claim 1 to further clarify features of the subject matter. **Independent Claim 1** as amended now recites a method implemented on a computing device having instructions stored on a computer-readable storage media and executable by a processor, to estimate security requirements needed to execute a managed code for a developer prior to an actual execution of the managed code, comprising:

> simulating the execution of all calls from an assembly to another assembly for all execution paths of one or more assemblies in the managed code, wherein the assembly comprises one or more files versioned and deployed as a unit, wherein the managed code is a managed shared library or an executable, wherein all managed code is contained within the one or more assemblies, wherein the execution of each assembly is statically simulated without actually running a corresponding managed code to simulate all possible calls and corresponding flow of argument data;
>
> finding a set of required permissions for each execution path by one or more simulated stack walks that each include a plurality of the assemblies, wherein each call in each execution path has a corresponding permissions set, wherein each assembly has one or more execution paths representing a different data and a control flow, and wherein the simulated stack walk comprises:
>
> > entering an execution path corresponding to a static simulation of execution of the assembly;
> >
> > entering a public entry point of a method in the assembly;
> >
> > gathering a permission set for the method in the assembly;
> >
> > determining whether the method in the assembly calls another method in the assembly or in an another assembly;
> >
> > gathering a permission set for the another method called by the method in the assembly; and
> >
> > creating a union of the gathered permission sets; and
>
> deriving the security requirements for execution paths corresponding to the one or more assemblies by using the union of the gathered permission sets across the execution paths corresponding to the one or more assemblies, wherein the union estimates the security requirements that will be triggered against the one or more assemblies during the actual execution of the one or more assemblies and whether a security exception will be triggered during the actual execution.

Applicant respectfully submits that no such method is disclosed, taught, or suggested by Koved and Scheifler, alone or in combination.

Koved and Scheifler Fail to Disclose, Teach, or Suggest Statically Simulating Execution of Each Assembly Through a Simulated Stack Walk to Derive the Security Requirements for Execution Paths Corresponding to one or more Assemblies

Estimating the Security Requirements that will be Triggered During the Actual Execution

Koved is directed towards determination of access rights needed to execute code (Koved, Abstract; § 1. Introduction). In Koved, access is automatically determined by using a modified interprocedural invocation graph called Access Rights Invocation Graph (Koved, § 1. Introduction). Koved takes advantage of Java's Permission Class hierarchy and uses the graph to propagate the access rights (Koved, § 1. Introduction).

Scheifler fails to compensate for the deficiencies of Koved. Scheifler is directed towards systems and methods that control access to a resource based on the source of the code and the identity of the principal on whose behalf the code is being executed (Scheifler, paras. 0001, 0015, 0026). Scheifler regulates access to resources requested by an operation executing on a computer (Scheifler, para. 0016). Scheifler utilizes a call stack for hierarchy of methods invoked by a thread (Scheifler, para. 0085). However, it seems that Scheifler does not simulate the execution, but instead applies the stack control during the execution (*see* Scheifler, paras. 0016, 0091).

Koved and Scheifler, alone or in combination, fail to disclose, teach, or suggest, "*simulating the execution of all calls from an assembly to another assembly for all execution paths of one or more assemblies in the managed code . . . wherein the execution of each assembly is statically simulated without actually running a corresponding managed code to simulate all possible calls and corresponding flow of argument data*" and "*deriving the security requirements for execution paths corresponding to the one or more assemblies by using the union of the gathered permission sets across the execution paths corresponding to the one or more assemblies,* *wherein the union estimates the security requirements that will be triggered against the one or more assemblies during the actual execution of the one or more assemblies and whether a security exception will be triggered during the actual execution*", as recited in Applicant's amended Claim 1.

Koved does not discuss or even mention a simulated stack walk, as recited in Applicant's amended Claim 1. Scheifler mentions utilizing a call stack, but Scheifler does not statically simulate a stack walk, as recited in Applicant's amended Claim 1, but rather applies a stack control during the actual execution. It follows that neither Koved or Scheifler discuss or even mention statically simulating execution of each assembly through a simulated stack walk, as recited in Applicant's amended Claim 1.

Applicant reviews the evidence and respectfully submits that the evidence no longer supports an obviousness rejection as Koved and Scheifler, alone or in combination, fail to disclose, teach, or suggest every feature recited in Applicant's amended Claim 1. Accordingly, Applicant respectfully requests that the § 103 rejection be withdrawn.

Independent Claims 12, 19, 26, and 38

Independent Claims 12, 19, 26, and 38 are directed to a method, a computer-readable storage media, an apparatus, and a computing device, respectively. These claims are allowable for reasons similar to those discussed above with respect to Claim 1, namely recitation of statically simulating execution of each assembly through a simulated stack walk to derive the security requirements for execution paths corresponding to the assemblies during an actual execution.

Independent Claim 12 recites in a managed code environment, a method implemented on a computing device having instructions stored on a computer-readable storage media and executable by a processor, comprising:

> simulating calling from one assembly to another for which a permission set is required, wherein the simulation comprises one or more simulated stack walks that include two or more of the assemblies, each assembly being managed code in a library, wherein an execution of each assembly is statically simulated without actually running a corresponding managed code to simulate all possible calls and corresponding flow of argument data, and wherein the simulated stack walk comprises:
>
> entering a public entry point of a method in the assembly;
>
> gathering a permission set for the method in the assembly;
>
> determining whether the method in the assembly calls another method in the assembly or in an another assembly;
>
> for each called method:
>
> gathering a permission set for the another method called by the method in the assembly; and
>
> determining whether the another method calls a subsequent method in the assembly or in the another assembly; and
>
> creating a union of the gathered permission sets;
>
> repeating the calling for each assembly in the managed code and for all possible execution paths of the managed code;
>
> repeating the entering for each public entry point in the library;
>
> finding the union of the permission sets corresponding to each call; and
>
> deriving security requirements for execution paths corresponding to the assemblies by using the union of the gathered permission sets across the execution paths corresponding to the one or more assemblies, wherein the union estimates the security requirements that will be triggered against the assemblies during an actual execution of the assemblies and whether a security exception will be triggered during the actual execution.

Applicant respectfully submits that Koved and Scheifler, alone or in combination, fail to disclose, teach, or suggest each and every feature of Claim 12, as amended. Accordingly, Applicant respectfully requests that the § 103 rejection be withdrawn.

Independent Claim 19 recites one or more computer-readable storage media having a tangible component comprising instructions that, when executed by a processor, perform a simulation of an execution of every data and control flow for managed code from which an estimate is derived of the minimum security requirements needed to dynamically execute the managed code without triggering a security exception, the instructions comprising:

> simulating one or more stack walks for each data and a control flow for the managed code, wherein the managed code corresponds to one or more assemblies, wherein the one or more stack walks comprise two or more of the assemblies, wherein the managed code makes use of a common language runtime (CLR) that is loaded upon the first invocation of a routine, and wherein the simulated stack walk comprises:
>
> entering a public entry point of a method in an assembly;
> gathering a permission set for the method;
> determining whether the method calls another method;
> for each called method:
> gathering a permission set for the called method; and
> determining whether the called method calls a subsequent method; and
> creating a union of the gathered permission sets; and
>
> deriving the security requirements for execution paths corresponding to the one or more assemblies by using the union of the gathered permission sets, wherein the union estimates the security requirements that will be triggered against the one or more assemblies during an actual execution of the one or more assemblies.

Applicant respectfully submits that Koved and Scheifler, alone or in combination, fail to disclose, teach, or suggest each and every feature of Claim 19, as amended. Accordingly, Applicant respectfully requests that the § 103 rejection be withdrawn.

Independent Claim 26 recites an apparatus comprising:

> means for processing;
>
> means for storing information in memory coupled to the means for processing;
>
> virtual machine means, stored in the memory, in a managed code portion, for operating a plurality of assemblies in managed code, wherein the managed code is a managed shared library or an executable and is in the managed code portion;
>
> execution engine means, in a native code portion, for executing the virtual machine means;

means, in the native code portion, for providing an operating system;

means for making a call in the managed code portion for access by one assembly to another assembly for which a permissions set is required;

means in the managed code portion for gathering the permissions set from each call;

means in the managed code portion for deriving a union of the gathered permissions sets;

means in the managed code portion for statically simulating the execution of all possible execution paths for the managed shared library or the executable without actually running a corresponding managed code, to derive therefrom the derived union of the gathered permissions sets wherein the means for simulating the execution performs, for each execution path, one or more simulated stack walks that each include a plurality of assemblies, and wherein the one or more simulated stack walks comprise:

means for entering a public entry point of a method in the assembly;

means for gathering a permission set for the method;

means for determining whether the method calls another method;

for each called method:

means for gathering a permission set for the called method;

means for determining whether the called method calls a subsequent method; and

means for repeating the previous gathering and determining until any gathered permission set is duplicative; and

means for creating a union of the gathered permission sets; and

means for deriving security requirements for execution paths corresponding to the plurality of assemblies by using the union of the gathered permission sets across the execution paths corresponding to the plurality of assemblies, wherein the union estimates whether a security exception will be triggered during an actual execution of the assemblies.

Applicant respectfully submits that Koved and Scheifler, alone or in combination, fail to disclose, teach, or suggest each and every feature of Claim 26, as amended. Accordingly, Applicant respectfully requests that the § 103 rejection be withdrawn.

Independent Claim 38 recites a computing device comprising:

a processor;

a memory coupled to the processor;

a managed code portion stored in the memory including a plurality of assemblies each being managed code in a managed shared library or in an executable;

a native code portion stored in the memory including:

an execution engine ; and

an operating system under the execution engine;

a virtual machine interfaced between the managed code portion and the native code portion and executed by the execution engine;

an application program in the managed code portion comprising logic configured to:

statically simulate the execution of all possible calls from one assembly to another for all possible execution paths of the managed code without actually running a corresponding managed code to simulate all possible calls and corresponding flow of argument data, wherein each assembly call has a corresponding permissions set, wherein the simulation of the execution comprises one or more simulated stack walks that each include a plurality of the assemblies, and wherein the one or more simulated stack walks comprise:

a public entry point of a method in the assembly;

a permission set for the method;

a determination of whether the method calls another method;

for each called method:

a permission set for the called method;

a determination of whether the called method calls a subsequent method; and

a totality of permission sets such that any subsequent permission set is duplicative; and

a union of the permission sets;

derive a union of the permissions sets from each assembly call; and

derivive security requirements for execution paths corresponding to the plurality of assemblies by using the union of the permission sets across the execution paths corresponding to the plurality of assemblies, wherein the union estimates the security requirements that will be triggered against the one or more assemblies during an actual execution of the assemblies.

Applicant respectfully submits that Koved and Scheifler, alone or in combination, fail to disclose, teach, or suggest each and every feature of Claim 38, as amended. Accordingly, Applicant respectfully requests that the § 103 rejection be withdrawn.

Dependent Claims 2-9, 11, 13-15, 17-18, 20-22, 24-25, 27-34, 36-37, 39-43, 45, and 51 depend directly or indirectly from one of independent Claims 1, 12, 19, 26, and 38, respectively, and are allowable by virtue of this dependency. These claims are also allowable for their own recited features that, in combination with those recited in independent Claims 1, 12, 19, 26, and 38 are not disclosed, taught, or suggested by Koved and Scheifler, alone or in combination.

Applicant respectfully submits that Koved and Scheifler, alone or in combination, do not render the claimed subject matter obvious and that the claimed subject matter, therefore, is patentably distinguishable over the cited references. For all of these reasons, Applicant respectfully request the §103(a) rejection of these claims be withdrawn.

CONCLUSION

Claims 1-9, 11-15, 17-22, 24-34, 36-43, 45, and 51 are in condition for allowance. Applicant respectfully requests reconsideration and prompt allowance of the subject application. If any issue remains unresolved that would prevent allowance of this case, the Office is requested to contact the undersigned attorney to resolve the issue.

Respectfully submitted,

Lee & Hayes, PLLC

Dated: 04/06/2009

By: / Dino Kujundzic /
Dino Kujundzic
Reg. No. 63,104
509-944-4762

Shirley L. Anderson
Reg. No. 57,763
509-944-4758